Decision Tree View of Neural Networks and Information Flow

1. Introduction

ReLU neural networks can be interpreted as **implicit decision trees** in which each input follows a path determined by activation patterns. Unlike classical decision trees, however, the nodes do not have direct access to the original input. Instead, each decision is made on a **transformed and partially filtered representation** of that input.

This difference creates a fundamental requirement:

> **Neural networks require low information loss across layers to preserve both decision capacity and compositional computation.**

---

2. Decision Trees vs Neural Networks

Classical Decision Trees

In a standard decision tree:

- Every node can directly access the **original input (x)**.
- Decisions at any depth can depend on **any feature**, regardless of earlier splits.
- Information is never destroyed—only **partitioned**.

ReLU Neural Networks

In a ReLU network:

$x_{k+1} = D_k W_k x_k$

- Each layer only sees the current state (x_k), not the original input.
- The original input is accessible **only through prior transformations**.
- Gating and linear mappings can **destroy or compress information**.

Thus:

> A neural network is a decision tree where each node sees only a *filtered memory* of the input.

---

3. Information Flow as a Structural Requirement

3.1 Access to Original Input

For a network to match the flexibility of a decision tree:

- Later layers must still be able to base decisions on **any relevant aspect of the original input**.

However:

- Once information is lost in early layers, it cannot be recovered.
- This restricts which branches of the implicit tree can exist.

Therefore:

> **Preserving input information is necessary to maintain the full branching capacity of the decision tree.**

3.2 Tolerance to Mixing vs Loss

A key distinction:

- **Mixing information** (e.g., via linear transforms) is acceptable.
- **Destroying information** (e.g., rank collapse, excessive sparsity) is not.

So:

> It does not matter how much the input is transformed, as long as it remains recoverable from the representation.

This aligns with distributed representations:

- Information can be spread across many coordinates
- Decisions can still access it if it has not been eliminated

---

4. Information Flow as Computational State

Beyond decision-making, the forward signal also carries **partial results**.

From the synthetic matrix view:

$y = M_\sigma x = W_L D_{L-1} \cdots W_1 x$

Each layer contributes to constructing the final transformation.

Thus:

- x_k encodes **intermediate computations**
- Later layers must build on what has already been formed

If this information is lost:

- the compositional structure breaks
- later layers cannot refine or complete earlier work

---

5. Hierarchical Associative Memory Perspective

Viewing the network as **hierarchical associative memory**:

- Each layer **recalls and constructs features** from the input
- These features are progressively refined and recombined

This introduces a second requirement:

> **Information that has already been recalled must be preserved and passed forward.**

Otherwise:

- previously extracted structure is discarded
- deeper layers cannot build higher-level associations

---

6. Dual Role of the Representation

At each layer, the state x_k serves two roles:

(A) Decision Context

- Determines the next gating matrix D_k
- Controls traversal through the implicit tree

(B) Computational Memory

- Stores intermediate results of the final mapping
- Enables compositional construction of output

Key Consequence

> Any loss of information harms both routing (decision-making) and computation simultaneously.

---

7. Effects of Information Loss

Early Information Loss

- Reduces distinguishability between inputs
- Forces different inputs onto the same tree path
- Decreases effective branching factor

Loss of Recalled Features

- Prevents accumulation of higher-level structure
- Forces later layers to recompute or approximate lost features
- Weakens hierarchical representation

Excessive Sparsity or Bottlenecks

- Limits accessible subspace of representations
- Reduces both decision diversity and mapping capacity

---

8. Why Some Information Loss is Tolerable

There is limited tolerance due to **distributed representations**:

- Natural data (e.g., images) is often sparse or structured
- Information spread across many dimensions is robust to partial loss
- Random or structured mixing can preserve recoverability

Thus:

> Moderate loss may be acceptable if the remaining representation still encodes the relevant information in a distributed form.

---

9. Synthesis

From the decision tree perspective:

- A classical tree retains full access to input at every node
- A neural network must *simulate this* through its internal representation

This leads to the central principle:

> **Unhindered information flow is necessary for a neural network to behave like a deep decision tree.**

It ensures that:

- future decisions can still depend on original input features
- intermediate computations can be composed into final outputs
- hierarchical feature construction is preserved

---

10. Conclusion

A ReLU neural network can be viewed as an implicit decision tree operating on a progressively transformed representation of the input. Unlike classical trees, it does not retain direct access to the input and must therefore preserve information through its layers.

> Information flow is not just beneficial but structurally necessary: it maintains both the decision-making capacity and the compositional computation that define the network's expressive power.

Reference

Neural networks as decision trees:

https://archive.org/details/decision-tree-re-lu